

magnum.fe: A micromagnetic finite-element simulation code based on FEniCS

Claas Abert^{*1,2}, Lukas Exl³, Florian Bruckner⁴, André Drews², and Dieter Suess⁴

¹*Fachbereich Mathematik, Universität Hamburg, Bundesstr. 55, D-20146 Hamburg, Germany*

²*Institut für Angewandte Physik und Zentrum für Mikrostrukturforschung, Universität Hamburg, Jungiusstr. 11, D-20355 Hamburg, Germany*

³*University of Applied Sciences, Department of Technology, A-3100 St.Poelten, Austria*

⁴*Institute of Solid State Physics, Vienna University of Technology, A-1040 Vienna, Austria*

January 16, 2013

Abstract

We have developed a finite-element micromagnetic simulation code based on the FEniCS package called magnum.fe. Here we describe the numerical methods that are applied as well as their implementation with FEniCS. We apply a transformation method for the solution of the demagnetization-field problem. A semi-implicit weak formulation is used for the integration of the Landau-Lifshitz-Gilbert equation. Numerical experiments show the validity of simulation results. magnum.fe is open source and well documented. The broad feature range of the FEniCS package makes magnum.fe a good choice for the implementation of novel micromagnetic finite-element algorithms.

Keywords: micromagnetics, finite-element method, Landau-Lifshitz-Gilbert equation

^{*}cabert@physnet.uni-hamburg.de

1 Introduction

Micromagnetic simulations are an important tool for the computational investigation of ferromagnetic materials. In recent years they were successfully used to describe magnetic effects ranging from permanent magnets to soft magnetic logic devices to magnetic recording structures [1, 2, 3, 4]. Many methods have been proposed to solve the micromagnetic equations numerically. Two popular approaches are the finite-difference method combined with the fast Fourier transform (FFT) [5, 6, 7] and the finite-element method combined with the boundary-element method [8, 9, 10]. Furthermore fast multipole methods [11, 12], nonuniform FFT [13] and low-rank tensor methods [14, 15] have been successfully applied to micromagnetic problems.

Different open-source codes for the finite-difference method [16, 17, 18] as well as for the finite-element method [19, 20] are available. Moreover there are a couple of reports on closed source simulation tools [9, 21, 22]. We present the open-source finite-element code `magnum.fe` that heavily relies on the recently published finite-element software FEniCS [23] and that solves the dynamic micromagnetic equations with a combination of two weak formulations.

Technically challenging tasks appearing in finite-element computations such as numbering of degrees of freedom, local to global mapping of cell integrals and numerical integration over tetrahedra are handled by FEniCS which offers a variety of finite-element bases including arbitrary order Lagrange elements and produces high performance code for the assembly of system matrices.

The high level of abstraction of FEniCS leads to a very concise code that naturally reflects the underlying numerical algorithms. This makes `magnum.fe` an ideal platform for the implementation of new micromagnetic finite-element algorithms. Implementing alternative weak formulations for certain subproblems can often be done with a few lines of Python.

This paper is structured as follows. In Sec. 2 we briefly present the theory of dynamical micromagnetism. In Sec. 3 we describe the numerical methods that are implemented in `magnum.fe`, namely a transformation method for the computation of the demagnetization field and a weak formulation for the integration of the Landau-Lifshitz-Gilbert equation as proposed in [24]. Section 4 gives an overview over the implementation of the algorithms and in Sec. 5 we show the validity of our code by means of numerical experiments.

2 Micromagnetism

Magnetization dynamics in the framework of micromagnetism are described by the Landau-Lifshitz-Gilbert equation (LLGE)

$$\partial_t \mathbf{m} = -\gamma(\mathbf{m} \times \mathbf{H}_{\text{eff}}) + \alpha(\mathbf{m} \times \partial_t \mathbf{m}) \quad (1)$$

where γ is the gyromagnetic ratio and $\alpha \geq 0$ is a phenomenological damping constant that depends on the material. The magnetization field \mathbf{m} is defined on a domain Ω and assumed to be normalized everywhere

$$|\mathbf{m}(\mathbf{x})| = 1, \quad \mathbf{x} \in \Omega. \quad (2)$$

This property is obviously preserved by the LLGE (1). The effective field \mathbf{H}_{eff} is given by the negative variational derivative of the Gibbs free energy $U(\mathbf{m})$ with respect to the magnetization \mathbf{m}

$$\mathbf{H}_{\text{eff}} = -\frac{1}{\mu_0 M_s} \frac{\delta U(\mathbf{m})}{\delta \mathbf{m}}. \quad (3)$$

The total effective field \mathbf{H}_{eff} is the sum of multiple contributions to the Gibbs free energy

$$\mathbf{H}_{\text{eff}} = \mathbf{H}_{\text{ex}} + \mathbf{H}_{\text{demag}} + \mathbf{H}_{\text{zeeman}} \quad (4)$$

where \mathbf{H}_{ex} is the exchange field, $\mathbf{H}_{\text{demag}}$ is the demagnetization field and $\mathbf{H}_{\text{zeeman}}$ is a constant external Zeeman field.

The exchange field \mathbf{H}_{ex} models the quantummechanical effect of the exchange interaction and is given by

$$\mathbf{H}_{\text{ex}} = \frac{2A_{\text{ex}}}{\mu_0 M_s} \Delta \mathbf{m} \quad (5)$$

where A_{ex} is the exchange constant and M_s is the saturation magnetization. Including the exchange field in the LLGE (1) gives rise to a boundary condition posed on the magnetization \mathbf{m}

$$\partial_{\boldsymbol{\nu}} \mathbf{m}(\mathbf{x}) = 0, \quad \mathbf{x} \in \partial\Omega \quad (6)$$

where $\partial_{\boldsymbol{\nu}}$ is the normal derivative. This boundary condition is often referred to as Brown condition [25].

The demagnetization field $\mathbf{H}_{\text{demag}}$ accounts for the magnetic dipole–dipole interaction. In the absence of electric current, the demagnetization field is curl-free and hence can be expressed as the gradient of a scalar potential u

$$\mathbf{H}_{\text{demag}} = -\nabla u. \quad (7)$$

The magnetic scalar potential u itself is the solution of a Poisson problem

$$\Delta u = M_s \nabla \cdot \mathbf{m}. \quad (8)$$

Boundary conditions for this Poisson problem are given as zero at infinity which is often referred to as open boundary conditions. With the Green's function of the Laplace operator the solution to (8) can be written in the integral form

$$u(\mathbf{x}) = \frac{M_s}{4\pi} \int \mathbf{m}(\mathbf{x}') \cdot \frac{\mathbf{x} - \mathbf{x}'}{|\mathbf{x} - \mathbf{x}'|^3} d\mathbf{x}' \quad (9)$$

which directly fullfills the open boundary conditions, see [6].

3 Numerics

3.1 Demagnetization Field

On a finite region Ω the Poisson equation (8) with Dirichlet boundary conditions is solved by the weak formulation

$$\int_{\Omega} \nabla u \cdot \nabla v \, d\mathbf{x} = M_s \int_{\Omega_{\text{sample}}} \mathbf{m} \cdot \nabla v \, d\mathbf{x} \quad \forall \quad v \in V \subset H^1. \quad (10)$$

where the boundary conditions are embedded in the function space V . However the problem under consideration has open boundary conditions. Consequently we would have to carry out the integration in (10) over the whole space which is not possible with the finite-element method.

In order to avoid this restriction we use a method called parallelepipedic shell transformation [26]. A finite cuboid shell Ω_{shell} is mapped onto the infinite exterior region $\Omega_{\text{ext}} \in \mathbb{R}^3 \setminus \Omega_{\text{cuboid}}$ via a bijective transformation $X(\mathbf{x})$

$$X : \Omega_{\text{shell}} \rightarrow \Omega_{\text{ext}}. \quad (11)$$

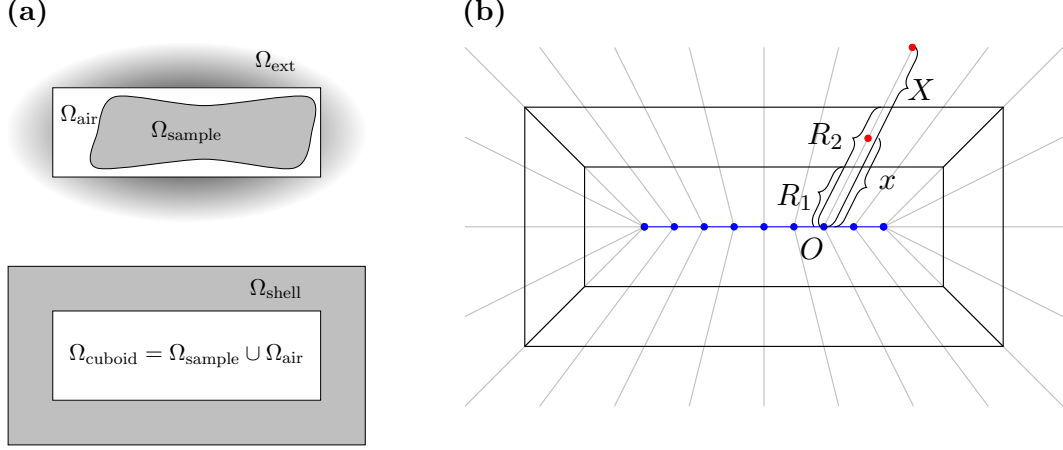


Figure 1: The transformation applied for the demagnetization-field computation. (a) The regions in transformed and untransformed space. (b) Sketch of the transformation showing the construction of the transformation direction. The transformation rays and origins marked by gray lines and blue points illustrate the construction of the transformation direction for different points in the shell.

The integration over $\Omega_{\text{ext}} = X(\Omega_{\text{shell}})$ and Ω_{shell} are connected via substitution by

$$\int_{\Omega_{\text{ext}}} f(x(X)) \, dX = \int_{\Omega_{\text{shell}}} f(x) |\det(DX(x))| \, dx. \quad (12)$$

When applying this to the left-hand side of (10) the gradients would still be calculated with respect to the shell coordinates x instead of the exterior coordinates X . The gradient with respect to exterior coordinates X is given by

$$[\nabla_X v](x) = [J^{-1} \nabla_x v](x) \quad (13)$$

with $J = DX(x)$ being the Jacobian of the transformation. Thus the weak formulation with shell transformation reads

$$\int_{\Omega_{\text{cuboid}}} \nabla u \cdot \nabla v \, d\mathbf{x} + \int_{\Omega_{\text{shell}}} (\nabla u)^T \mathbf{g} \nabla v \, d\mathbf{x} = M_s \int_{\Omega_{\text{sample}}} \mathbf{m} \cdot \nabla v \, d\mathbf{x} \quad \forall \quad v \in V \quad (14)$$

with the metric tensor \mathbf{g} given by

$$\mathbf{g} = J^{-1,T} J^{-1} |\det J|. \quad (15)$$

The Dirichlet boundary condition $u = 0$ on $\partial(\Omega_{\text{cuboid}} \cup \Omega_{\text{shell}})$ is embedded in the function space V . The metric tensor \mathbf{g} is symmetric positive definite, hence the symmetric bilinear form on the left-hand side of (14) is also positive definite. Thus, by the right choice of the subspace $V \subset H^1$, problem (14) has a unique solution.

3.1.1 Choice of Transformation

We choose the shape of the shell to be cuboid as described in [26]. The transformation from Ω_{shell} to Ω_{ext} is carried out along rays as sketched in Fig. 1b. In the simple case of a cubic sample the transformation has a fixed origin and is radial. However in the general case the origin has

to be variable in order to obtain a continuous transformation across shell-patch borders. The transformation origin in this case moves on the middle plane that is perpendicular to the shortest edge of the cuboid.

In the following the properties of the one-dimensional transformation in the radial directions are discussed. Obviously there are many possible choices for such one-dimensional transformations that are bijective and fullfill (11). A suitable transformation distorts the basis functions used for discretization in a way that the decay of the potential u may be approximated accurately. As can be seen in (9) the decay of the potential is $u \propto 1/|\mathbf{x}|^2$ in the far-field approximation. In the following piecewise linear basis functions

$$\phi(x) = a + bx \quad (16)$$

are considered in the untransformed space. In order to obtain a test function decaying with $1/X^2$ the transformation $X : \Omega_{\text{shell}} \rightarrow \Omega_{\text{ext}}$ has to fullfill

$$x = a' + b' \frac{1}{X^2} \quad (17)$$

and thus

$$X = \sqrt{\frac{b'}{x - a'}} \quad (18)$$

Furthermore the transformation X has to map the finite interval $[R_1, R_2]$ to the infinite interval $[R_1, \infty]$, see Fig. 1b.

$$X(R_1) = \sqrt{\frac{b'}{R_1 - a'}} = R_1 \quad (19)$$

$$1/X(R_2) = \sqrt{\frac{R_2 - a'}{b'}} = 0 \quad (20)$$

This immediatly yields

$$X = R_1 \sqrt{\frac{R_2 - R_1}{R_2 - x}} \quad (21)$$

as suitable transformation for linear basis functions.

When using higher order polynomials as basis functions we use the transformation

$$X = R_1 \frac{R_2 - R_1}{R_2 - x}. \quad (22)$$

instead of (21). This transforms second and third order polynomials like

$$a + bx + cx^2 \rightarrow a' + b' \frac{1}{X} + c' \frac{1}{X^2} \quad (23)$$

$$a + bx + cx^2 + dx^3 \rightarrow a' + b' \frac{1}{X} + c' \frac{1}{X^2} + d' \frac{1}{X^3} \quad (24)$$

which enables a much better approximation of the decaying magnetic potential.

3.2 Landau-Lifshitz-Gilbert Equation

We solve the Landau-Lifshitz-Gilbert equation (1) with a weak formulation originally proposed by Alouges in [24]. In a first step we set $\mathbf{v} = \partial_t \mathbf{m}$ and multiply with vector test functions which yields

$$\int_{\Omega} (\mathbf{v} - \alpha \mathbf{m} \times \mathbf{v}) \cdot \boldsymbol{\phi} \, d\mathbf{x} + \int_{\Omega} \gamma (\mathbf{m} \times \mathbf{H}_{\text{eff}}) \cdot \boldsymbol{\phi} \, d\mathbf{x} = 0 \quad \forall \quad \boldsymbol{\phi} \in V^3. \quad (25)$$

The terms of the right-hand side of the LLGE (1) and thus also the left-hand side are perpendicular to the magnetization \mathbf{m} . Therefore it is sufficient to test the equation with test functions $\phi \in T_{\mathbf{m}}$ and restrict the solution space of \mathbf{v} to $T_{\mathbf{m}}$ where $T_{\mathbf{m}}$ is the tangent space to the magnetization \mathbf{m} . Following Alouges we set $\phi = \mathbf{m} \times \mathbf{w}$ in (25) and restrict the new test functions \mathbf{w} to the tangent space $T_{\mathbf{m}}$, which yields

$$\int_{\Omega} (\alpha \mathbf{v} + \mathbf{m} \times \mathbf{v}) \cdot \mathbf{w} \, d\mathbf{x} - \gamma \int_{\Omega} \mathbf{H}_{\text{eff}}(\mathbf{m}) \cdot \mathbf{w} \, d\mathbf{x} = 0 \quad \forall \quad \mathbf{w} \in T_{\mathbf{m}}. \quad (26)$$

This scheme can be extended to an implicit θ -scheme by replacing \mathbf{m} with $\mathbf{m} + \theta k \mathbf{v}$ with the timestep k and $\theta \in [0, 1]$

$$\int_{\Omega} (\alpha \mathbf{v} + \mathbf{m} \times \mathbf{v}) \cdot \mathbf{w} \, d\mathbf{x} - \gamma \int_{\Omega} \mathbf{H}_{\text{eff}}(\mathbf{m} + \theta k \mathbf{v}) \cdot \mathbf{w} \, d\mathbf{x} = 0 \quad \forall \quad \mathbf{w} \in T_{\mathbf{m}}. \quad (27)$$

In contrast to (25) effective field terms which are linear in the magnetization can be integrated implicitly without breaking the linearity of the scheme. E.g. considering only the exchange field (5) and performing integration by parts yields

$$\int_{\Omega} (\alpha \mathbf{v} + \mathbf{m} \times \mathbf{v}) \cdot \mathbf{w} \, d\mathbf{x} + \frac{2A_{\text{ex}}\gamma}{\mu_0 M_s} \int_{\Omega} \nabla(\mathbf{m} + \theta k \mathbf{v}) \cdot \nabla \mathbf{w} \, d\mathbf{x} = 0 \quad \forall \quad \mathbf{w} \in T_{\mathbf{m}} \quad (28)$$

where the boundary condition (6) has been taken into account. When using the whole effective field (4) it is sufficient to treat \mathbf{H}_{ex} implicitly in order to get a stable scheme [13]. This is of special practical importance since the calculation of the demagnetization field $\mathbf{H}_{\text{demag}}$ is very time-consuming since the corresponding discretized operators are dense and thus in general not feasible to be computed. We therefore calculate the demagnetization potential u as described in Sec. 3.1 and treat it only explicitly. Including \mathbf{H}_{ex} implicitly and $\mathbf{H}_{\text{demag}}$ and $\mathbf{H}_{\text{zeeman}}$ explicitly yields the weak formulation

$$\begin{aligned} \int_{\Omega} (\alpha \mathbf{v} + \mathbf{m} \times \mathbf{v}) \cdot \mathbf{w} \, d\mathbf{x} + \frac{2A_{\text{ex}}\gamma}{\mu_0 M_s} \int_{\Omega} \theta k \nabla \mathbf{v} \cdot \nabla \mathbf{w} \, d\mathbf{x} \\ = \gamma \int_{\Omega} (\mathbf{H}_{\text{zeeman}} - \nabla u) \cdot \mathbf{w} \, d\mathbf{x} - \frac{2A_{\text{ex}}\gamma}{\mu_0 M_s} \int_{\Omega} \nabla \mathbf{m} \cdot \nabla \mathbf{w} \, d\mathbf{x} \quad \forall \quad \mathbf{w} \in T_{\mathbf{m}} \end{aligned} \quad (29)$$

which can be written as

$$a(\mathbf{v}, \mathbf{w}) = L(\mathbf{w}) \quad \forall \quad \mathbf{w} \in T_{\mathbf{m}}. \quad (30)$$

where $a(\mathbf{v}, \mathbf{w})$ is a bilinear form and $L(\mathbf{w})$ is a linear form. As shown in [24] the bilinear form $a(\mathbf{v}, \mathbf{w})$ can be written as the sum of a skew-symmetric form and a symmetric positive definite form. Thus the problem (29) possesses a unique solution \mathbf{v} .

In order to relieve the tangent-space constraint on the test functions \mathbf{w} , (29) has to be supplemented by a term that accounts for the part of \mathbf{w} parallel to the magnetization \mathbf{m} . Together with the tangent-space constraint for the solution \mathbf{v} the system then reads

$$a(\mathbf{v}, \mathbf{w}) + \int_{\Omega} \lambda \mathbf{m} \cdot \mathbf{w} \, d\mathbf{x} = L(\mathbf{w}) \quad \forall \quad \mathbf{w} \in V^3 \quad (31)$$

$$\int_{\Omega} \mathbf{v} \cdot \mathbf{m} \, d\mathbf{x} = 0 \quad \forall \quad \sigma \in V. \quad (32)$$

If (\mathbf{v}, λ) is a solution to this system then \mathbf{v} solves (29). We discretize (31) and the constraint (32) choosing the same order of finite elements for the scalar field λ as for the solution \mathbf{v} . This leads to a saddle-point problem of the form

$$\begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ \lambda \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix} \quad (33)$$

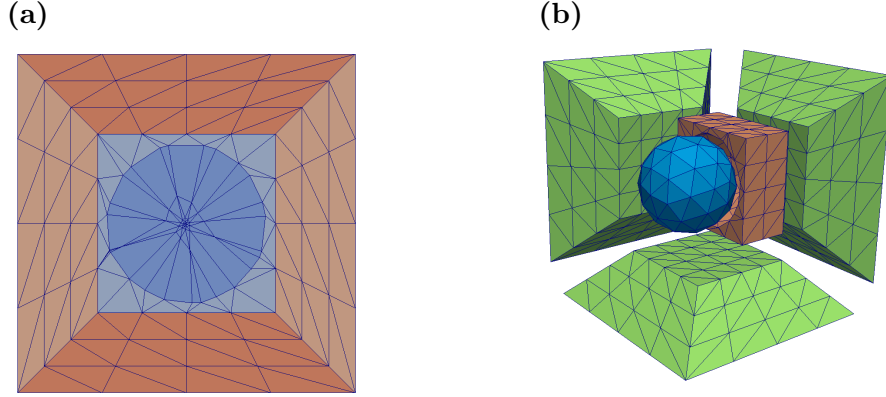


Figure 2: Mesh of a spherical sample including the cuboid shell for the demagnetization-field computation generated with magnum.fe using Gmsh. (a) Cut through the middle plane of the mesh. (b) Explosion view of the the meshed sphere including the cuboid shell patches.

where $A \in \mathbb{R}^{3N \times 3N}$ corresponds to the bilinear form $a(\mathbf{v}, \mathbf{w})$, $f \in \mathbb{R}^{3N}$ corresponds to the linear form $L(\mathbf{w})$ and $B \in \mathbb{R}^{3N \times N}$ and B^T correspond to the integrals in (31) and (32). Since B has full rank and A is regular, the system (33) has a unique solution.

A single integration step is carried out by first solving (33) for a given magnetization $\mathbf{m}(t)$ and then proceed in time by

$$\mathbf{m}_i(t+k) = \frac{\mathbf{m}_i(t) + k\mathbf{v}_i}{|\mathbf{m}_i(t) + k\mathbf{v}_i|} \quad (34)$$

where \mathbf{m}_i and \mathbf{v}_i are the nodal values at node i of the discretized magnetization \mathbf{m} and solution \mathbf{v} .

4 Implementation

Finite element software basically has to solve three sub problems: Mesh generation, system-matrix assembly and solution of the resulting linear systems of equations. We use Gmsh [27] for mesh generation and FEniCS [23] for matrix assembly and the solution of linear systems.

For the automated generation of a suitable mesh for the demagnetization-field computation we use the C++ interface of Gmsh. magnum.fe is able to produce regular meshes for rectangular samples or alternatively read mesh information from a mesh file, which may implement any format supported by Gmsh. Then a cuboid shell is wrapped around the sample and meshed with Gmsh, see Fig. 2. The number of shell layers which largely influences the quality of the demagnetization-field approximation is configurable.

The system-matrix assembly is done with FEniCS which offers a C++ interface as well as a Python interface. For magnum.fe we mainly use the Python interface which leads to a very concise code that represents the mathematical problem at hand very naturally. Listing 1 and 2 show excerpts of the magnum.fe code, namely the definition of the weak formulations (14) and (31)-(32) with the unified form language (UFL)[28] defined by FEniCS. From these form definitions FEniCS creates and compiles a fast C++ code for the matrix assembly via the FEniCS form compiler (FFC)[29]. Thus a high performance is achieved although the actual programming is done in the scripting language Python.

However when hitting the limits of FEniCS it is often not possible to extend the functionality with Python without facing performance issues. All performance relevant extensions to FEniCS

```

1 # Set up mixed function space
2 VV = VectorFunctionSpace(mesh, "CG", 1) # Solution
3 VS = FunctionSpace(mesh, "CG", 1)      # Lagrange Multipliers
4 V = VV * VS
5
6 # Set up test and trial functions
7 (v, lamb) = TrialFunctions(V)
8 (w, sigma) = TestFunctions(V)
9
10 # Prefactor for exchange field
11 f_ex = (- 2.0 * Aex * gamma) / (mu0 * ms)
12
13 # Set up bilinear form
14 a = alpha * dot(v, w) * dx \
15     + dot(cross(m, v), w) * dx \
16     + sigma * inner(m, v) * dx \
17     + lamb * inner(m, w) * dx \
18     - 0.5 * dt * f_ex * Dx(v[i],j) * Dx(w[i],j) * dx
19
20 # Set up linear form
21 L = f_ex * Dx(m[i],j) * Dx(w[i],j) * dx

```

Listing 1: Weak form of the Landau-Lifshitz-Gilbert equation as explained in Sec. 3.2. A mixed function space is used to include the solution \mathbf{v} and the scalar field λ (called lamb here, since lambda is a Python keyword).

```

1 # Set up the bilinear form for the demagnetization problem
2 a = Dx(v, i) * Dx(u, i) * dx(0) + \
3     Dx(v, i) * Dx(u, i) * dx(1) + \
4     Dx(v, i) * gx[i,j] * Dx(u, j) * dx(2) + \
5     Dx(v, i) * gy[i,j] * Dx(u, j) * dx(3) + \
6     Dx(v, i) * gz[i,j] * Dx(u, j) * dx(4)
7
8 # Create Dirichlet boundary conditions
9 bc = DirichletBC(VS, Constant(0.0), DomainBoundary())
10
11 # Use the SymmetricAssembler to assemble the system matrix
12 A, An = symmetric_assemble(a, bc)

```

Listing 2: Weak form of the demagnetization-field problem. The matrix-valued expressions \mathbf{gx} , \mathbf{gy} and \mathbf{gz} are multiplied using indices i , j and k and integrated over different subdomains of the mesh.

are thus written in C++. Like FEniCS we use SWIG [30] to exploit the interface of the extensions in Python. These extensions include the aforementioned Gmsh interface for the generation of the cuboid shells. Furthermore an extension for the assembly of matrices and vectors that arise from pointwise calculations of functions was written. This extension works for n th-order Lagrange elements and performs calculations on the (auxiliary-) nodes of the function space. It is used for the renormalization step (34) which is applied only on the nodes as well as for an alternative implementation of the extended system (31)-(32) where the constraint (2) is restricted to the nodes.

The algorithms presented in Sec. 3 are implemented with Lagrange functions, which are piecewise polynomial and globally continuous. In case of the demagnetization field the order of the elements is configurable. For the solution of the Landau-Lifshitz-Gilbert equation we choose 1st order elements. Assembly code generated by FEniCS uses Gauss quadrature for integration. The degree of quadrature is chosen according to the polynomial degree of the integrand and is thus exact for cell-wise polynomial functions. Analytical expressions are interpolated to a given degree before integration, e.g. we chose the metric tensor \mathbf{g} to be integrated 5th order.

For the solution of the resulting linear systems of equations FEniCS offers interfaces to a variety of open source linear algebra backends. magnum.fe uses the Trilinos Epetra [31] backend for both the demagnetization problem and the solution of the LLGE. Since in both cases the resulting system matrices are sparse, iterative Krylov-space methods are applied. The matrix of the demagnetization problem is symmetric and positive definite as shown in Sec. 3.1. Thus a conjugate gradient solver in combination with an algebraic multigrid preconditioner is used. The saddle-point problem arising from the Landau-Lifshitz-Gilbert equation is solved by an ILU preconditioned GMRES solver.

5 Numerical Experiments

5.1 Demagnetization Field

For validation and comparison of the demagnetization-field algorithm of different order as presented in Sec. 3.1 we choose a homogeneously magnetized unit cube. The energy of this system can be computed as

$$E = \frac{1}{2} \int_{\Omega} \nabla u \cdot \mathbf{m} \, dx \quad (35)$$

which is $1/6$ if $\mathbf{m} = (0, 0, 1)$. Figure 3 shows results for different polynomial degree of the basis functions and different number of non-zero system-matrix entries. The latter was chosen as measure since a matrix-vector multiplication is linear in this size and so is the iterative solution of the associated linear system. Also the storage requirements are linear in the number of non-zero matrix entries.

Figure 3a shows the results of the energy calculations. The 2nd and 3rd order method perform clearly better than the 1st order method for the same number of matrix entries. This is a consequence of the additional $1/X$ term that the higher order elements provide in contrast to the 1st order elements, see (23) and (24).

Figure 3b shows the number of iterations needed for the iterative solution of the linear system of equations. Together with the numerical complexity of a single matrix-vector multiplication, the number of iterations gives the over-all complexity of the demagnetization-field algorithm. The log-log plot yields a linear dependence with slope ≈ 0.4 , resulting in an over-all complexity of $\mathcal{O}(N^{1.4})$ for the demagnetization-field algorithm.

Finally Fig. 3c and Tab. 3d show the convergence rates of the algorithm. In order to account for auxiliary nodes for the higher order methods the discretization h is not taken from the mesh,

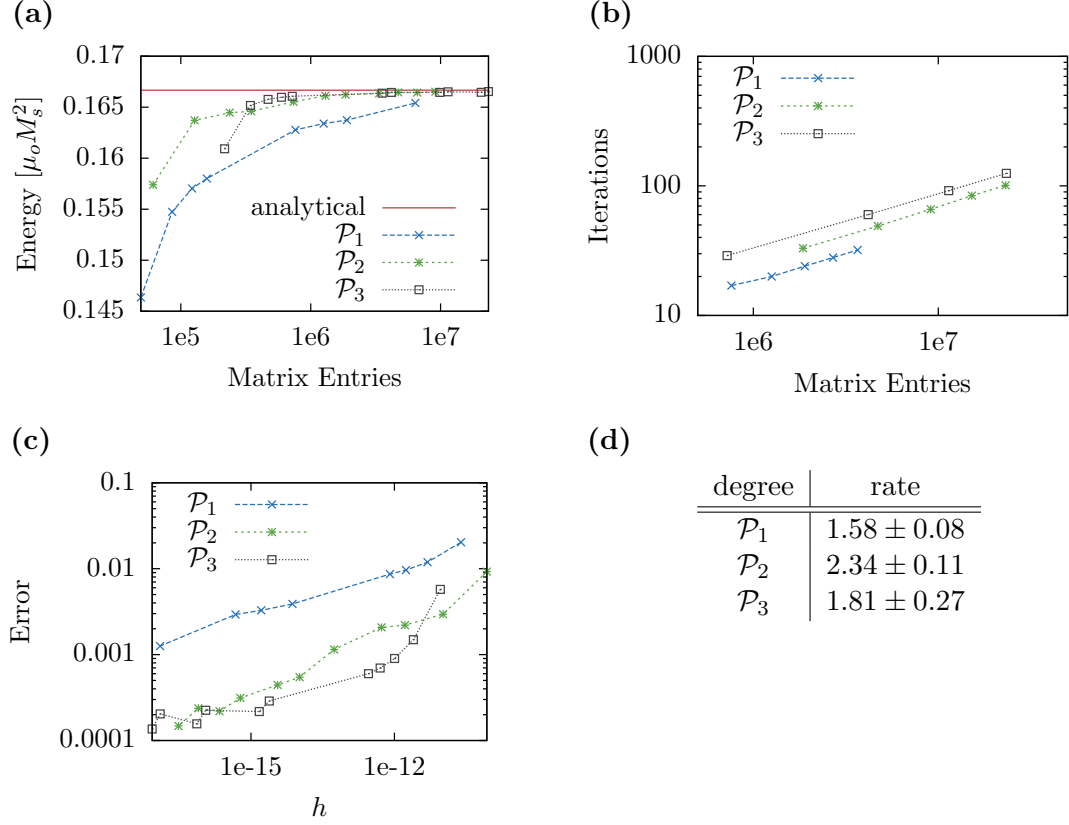


Figure 3: Demagnetization energy calculations for different order polynomial test and trial functions and different numbers of matrix entries. (a) Energy calculations for a homogeneously magnetized unit cube. (b) Iterations of the Conjugate Gradient solver. (c) and (d) Convergence rates for different order polynomials.

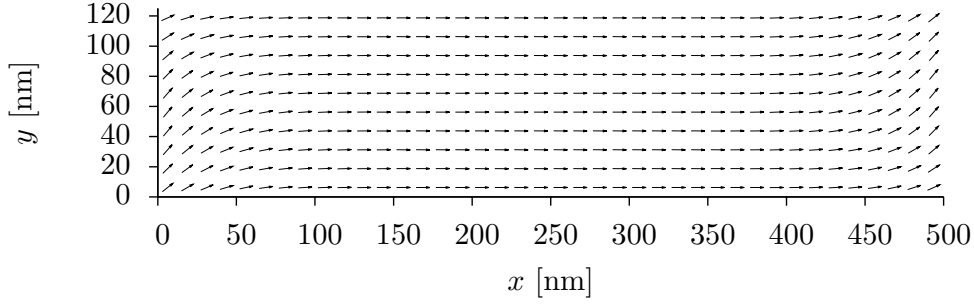


Figure 4: In-plane magnetization configuration of a so-called s-state in a rectangular thin film of the size $500 \times 125 \times 3\text{nm}^3$ with the material parameters of permalloy.

but estimated by $N^{-1/3}$ where N is the number of degrees of freedom, i.e. the total number of nodes and auxiliary nodes. Again the 1st order method shows the poorest performance. The 2nd and 3rd order methods have convergence rates of approximately 2 which is expected from other demagnetization methods, see [7].

For further numerical experiments with the presented demagnetization-field method and comparison to other recently developed methods see [7].

5.2 Landau-Lifshitz Equation

The method for the integration of the Landau-Lifshitz-Gilbert equation is validated with the standard problem #4 proposed by the Micromagnetic Modeling Activity Group μMAG [32]. A rectangular sample of the size $500 \times 125 \times 3\text{nm}^3$ is relaxed in a so-called s-state with the bulk magnetization pointing in the x -direction, see Fig. 4. The material parameters of the sample are chosen similar to those of Permalloy

$$A_{\text{ex}} = 1.3 \cdot 10^{-11} \text{ J/m} \quad (36)$$

$$M_{\text{s}} = 8.0 \cdot 10^5 \text{ A/m} \quad (37)$$

$$\alpha = 0.2. \quad (38)$$

Then, in addition to the exchange field and the demagnetization field, a homogeneous external Zeeman field $\mathbf{H}_{\text{zeeman}} = (-24.6, 4.3, 0)\text{mT}$ is applied which results in a switching process. Figure 5 shows the evolution of the averaged magnetization components in time as calculated by magnum.fe compared to the results of the finite-difference code MicroMagnum [18].

6 Conclusion and Outlook

We present the open-source micromagnetic software magnum.fe. magnum.fe is a complete three-dimensional finite-element code, which computes magnetization dynamics with a combination of two linear weak formulations. It is written in C++ and Python and uses the finite-element package FEniCS [23]. The correctness of the code is demonstrated by a number of numerical experiments including the μMAG standard problem #4.

Due to the multitude of features and the high level of abstraction of FEniCS, magnum.fe is well suited for the implementation of novel finite-element algorithms. magnum.fe itself is well documented and unit tested and is freely available at github [33].

We plan to implement alternative demagnetization-field algorithms as well as integration schemes for the Landau-Lifshitz-Gilbert equation. Contributions to magnum.fe are very welcome.

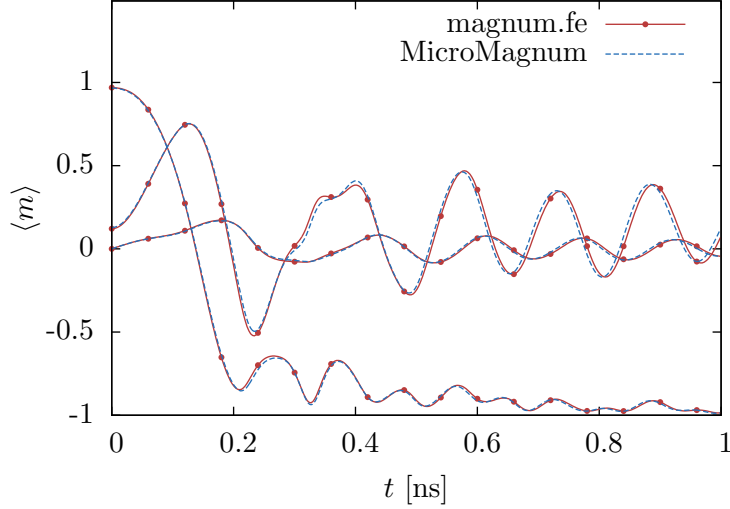


Figure 5: Time evolution of the averaged magnetization for the standard problem #4. The results of magnum.fe are compared to that of the finite-difference package MicroMagnum. The different branches correspond to the components of the averaged magnetization.

Acknowledgements

We thank Michael Hinze and Guido Meier for fruitful discussions. Financial support by the Deutsche Forschungsgemeinschaft via the Graduiertenkolleg 1286 “Functional Metal-Semiconductor Hybrid Systems”, the Austrian Science Fund (FWF, project SFB-ViCoM F4112-N13), and the Sonderforschungsbereich 668 “Magnetism from the single atom to the nanostructure” is gratefully acknowledged.

References

- [1] J. Zhu and H. N. Bertram, “Micromagnetic studies of thin metallic films (invited),” *J. Appl. Phys.*, vol. 63, pp. 3248–3253, apr 1988.
- [2] M. E. Schabes, “Micromagnetic theory of non-uniform magnetization processes in magnetic recording particles,” *J. Magn. Magn. Mater.*, vol. 95, no. 3, pp. 249–288, 1991.
- [3] R. Hertel, “Micromagnetic simulations of magnetostatically coupled nickel nanowires,” *J. Appl. Phys.*, vol. 90, pp. 5752–5758, dec 2001.
- [4] M. Kläui, C. A. F. Vaz, J. A. C. Bland, W. Wernsdorfer, G. Faini, E. Cambril, L. J. Heyderman, F. Nolting, and U. Rüdiger, “Controlled and reproducible domain wall displacement by current pulses injected into ferromagnetic ring structures,” *Phys. Rev. Lett.*, vol. 94, p. 106601, Mar 2005.
- [5] D. V. Berkov, K. Ramstreck, and A. Hubert, “Solving micromagnetic problems. towards an optimal numerical method,” *physica status solidi (a)*, vol. 137, no. 1, pp. 207–225, 1993.
- [6] C. Abert, G. Selke, B. Krüger, and A. Drews, “A fast finite-difference method for micromagnetics using the magnetic scalar potential,” *IEEE Trans. Magn.*, 2012.
- [7] C. Abert, L. Exl, G. Selke, A. Drews, and T. Schrefl, “Numerical methods for the stray-field calculation: A comparison of recently developed algorithms,” *J. Magn. Magn. Mater.*, vol. 326, no. 0, pp. 176–185, 2013.

- [8] J. Fidler and T. Schrefl, “Micromagnetic modelling - the current state of the art,” *J. Phys. D: Appl. Phys.*, vol. 33, no. 15, p. R135, 2000.
- [9] D. Suess, V. Tsiantos, T. Schrefl, J. Fidler, W. Scholz, H. Forster, R. Dittrich, and J. Miles, “Time resolved micromagnetics using a preconditioned time integration method,” *J. Magn. Magn. Mater.*, vol. 248, no. 2, pp. 298 – 311, 2002.
- [10] F. Bruckner, C. Vogler, M. Feischl, D. Praetorius, B. Bergmair, T. Huber, M. Fuger, and D. Suess, “3D FEM-BEM-coupling method to solve magnetostatic Maxwell equations,” *J. Magn. Magn. Mater.*, vol. 324, no. 10, pp. 1862 – 1866, 2012.
- [11] R. Beatson and L. Greengard, “A short course on fast multipole methods,” *Wavelets, Multilevel Methods and Elliptic PDEs*, pp. 1–37, 1997.
- [12] J. Blue and M. Scheinfein, “Using multipoles decreases computation time for magnetostatic self-energy,” *IEEE Trans. Magn.*, 1991.
- [13] F. Alouges, E. Kritsikis, and J.-C. Toussaint, “A convergent finite element approximation for Landau-Lifschitz-Gilbert equation,” *Physica B: Condensed Matter*, vol. 407, no. 9, pp. 1345 – 1349, 2012.
- [14] L. Exl, W. Auzinger, S. Bance, M. Gusenbauer, F. Reichel, and T. Schrefl, “Fast stray field computation on tensor grids,” *J. Comput. Phys.*, vol. 231, no. 7.
- [15] L. Exl, C. Abert, N. J. Mauser, T. Schrefl, H. P. Stimming, and D. Suess, “FFT-based Kronecker product approximation to micromagnetic long-range interactions,” 2012. arXiv:1212.3509.
- [16] M. Donahue and D. Porter, “OOMMF user’s guide, version 1.0,” *Interagency Report*, vol. NISTIR 6376, 1999.
- [17] A. Vansteenkiste and B. Van De Wiele, “MuMax: a new high-performance micromagnetic simulation tool,” *J. Magn. Magn. Mater.*, vol. 323, no. 21, pp. 2585–2591, 2011.
- [18] “MicroMagnum.” <http://micromagnum.informatik.uni-hamburg.de/>.
- [19] T. Fischbacher, M. Franchin, G. Bordignon, and H. Fangohr, “A systematic approach to multiphysics extensions of finite-element-based micromagnetic simulations: Nmag,” *Magnetics, IEEE Transactions on*, 2007.
- [20] W. Scholz, J. Fidler, T. Schrefl, D. Suess, R. Dittrich, H. Forster, and V. Tsiantos, “Scalable parallel micromagnetic solvers for magnetic nanostructures,” *Computational Materials Science*, vol. 28, no. 2, pp. 366 – 383, 2003.
- [21] A. Kakay, E. Westphal, and R. Hertel, “Speedup of fem micromagnetic simulations with graphical processing units,” *IEEE Trans. Magn.*, vol. 46, pp. 2303 –2306, june 2010.
- [22] R. Chang, S. Li, M. V. Lubarda, B. Livshitz, and V. Lomakin, “Fastmag: Fast micromagnetic simulator for complex magnetic structures (invited),” *J. Appl. Phys.*, vol. 109, pp. 07D358 –07D358–6, apr 2011.
- [23] A. Logg, K.-A. Mardal, G. N. Wells, *et al.*, *Automated Solution of Differential Equations by the Finite Element Method*. Springer, 2012.

- [24] F. Alouges, “A new finite element scheme for Landau–Lifchitz equations,” *Discrete Contin. Dyn. Syst. Ser. S*, vol. 1, no. 2, pp. 187–196, 2008.
- [25] W. F. Brown Jr., *Micromagnetics*. New York: Interscience Publisher, 1963.
- [26] X. Brunotte, G. Meunier, and J. Imhoff, “Finite element modeling of unbounded problems using transformations: a rigorous, powerful and easy solution,” *IEEE Trans. Magn.*, 1992.
- [27] C. Geuzaine and J. Remacle, “Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities,” *International Journal for Numerical Methods in Engineering*, vol. 79, no. 11, pp. 1309–1331, 2009.
- [28] M. S. Alnæs, *Automated Solution of Differential Equations by the Finite Element Method*, ch. 17. Springer, 2012.
- [29] R. C. Kirby and A. Logg, “A compiler for variational forms.” *ACM Trans. Math. Software* 32(3), 2006, pp. 417-444, 2011.
- [30] “SWIG.” <http://www.swig.org/>.
- [31] M. Heroux, R. Bartlett, V. H. R. Hoekstra, J. Hu, T. Kolda, R. Lehoucq, K. Long, R. Pawlowski, E. Phipps, A. Salinger, H. Thornquist, R. Tuminaro, J. Willenbring, and A. Williams, “An Overview of Trilinos,” Tech. Rep. SAND2003-2927, Sandia National Laboratories, 2003.
- [32] R. McMichael, “Standard problem number 4, problem specification and reported solutions,” *Micromagnetic Modeling Activity Group*, 1998.
- [33] “magnum.fe.” <http://github.com/c-abird/magnum.fe/>.